CRISPR Phylogeny Potential Approach

September 15, 2022

1 Methods

1.1 Mutation Model

Denote by N the number of cells sequenced following an CRISPR-Cas9 lineage tracing experiment. Within each cell is an recording cassette consisting of Σ sites. A site in the recording cassette may remain unedited, or may have contain a CRISPR-Cas9 induced mutation or deletion, either inherited from its ancestors or acquired *de novo*. We denote an unedited mutation state at site σ by ϕ_σ and index potential mutations at that site by $1, ..., M_\sigma$. If a site is deleted, we represent its mutation state by D.

For a given cell, c, denote the mutation state at site σ by $\sigma_c \in \{\phi_\sigma, 1, ..., M_\sigma, D\}$.

The accumulation of CRISPR-Cas9 induced mutations at a given site behaves as a time-homogeneous, continuous time Markov Chain. At a given site, transitions may occur from an unedited state to a mutated or deleted state, or from a mutated state to a deleted state, depending on a site-specific mutation rate, λ_s as well as a site specific transition probability matrix P^s. Therefore, the transition probability function, $P^s(t), t \geq 0$, is a row-stochastic matrix that defines a time-homogeneous and time-irreversible model of mutation. For any mutation states a, b, the entry $P^s_{ab}(t)$ is the conditional probability of observing state b at any site t time units after observing state a at that site. We denote by Q the infinitesimal generator of P, and by π, the stationary distribution.

We can compute the evolutionary distance between two cells i and j, given their respective vectors of mutation states at each site in the recording cassette. Given that the mutation model in this lineage tracing system is a time-irreversible model, the distance between cells i and j at a particular site is determined by the distance in transitioning from the ancestral state to both σ_i and σ_j respectively. We will make the further simplifying assumption that the ancestral state is equidistant to both of the cells. As such, the likelihood for a given pair of cells (i, j) is given by

$$\mathscr{L}_{i,j}(t) = \prod_{\text{sites } \sigma} \sum_{a \in A(\sigma)} \pi_a P_{a\sigma_i}(t/2) P_{a\sigma_j}(t/2)$$

where π_a is a prior over ancestral statesaand $A(\sigma)$ denotes the set of all possible ancestral states for cells i, j at site σ. We are rescued from excessive complexity by the biological constraints of the system, which lead to a very limited set of possible ancestral states at a given site for a pair of cells, consisting of the unedited stated and the least common ancestor (LCA) of the two. That is, if $\sigma_i = 1$ and $\sigma_j = 2$, then $\text{LCA}(1, 2) = 0$ and $A = \{0\}$. If $\sigma_i = 1$ and $\sigma_j = 1$, then $A = \{0, 1\}$, to account for the possibility of the same mutation recurring independently.

As such, the log-likelihood has the form

$$\log \mathscr{L}_{i,j}(t) = \sum_{\text{sites } \sigma} \log \sum_{a \in A} \pi_a P_{a\sigma_i}(t/2) P_{a\sigma_j}(t/2)$$

Over all pairs of cells (i, j), the log-likelihood is:

$$\log \mathscr{L}(t) = \sum_{i \neq j} \sum_{\text{sites } \sigma} \log \sum_{a \in A} \pi_a P_{a\sigma_i}(t/2) P_{a\sigma_j}(t/2)$$

1.2 Derivation of Gradient Update Step

In the previous section, we derived an expression for the log-likelihood of the data as a function of time, t. In a phylogenetic tree, this time corresponding to tree distances between leaves. Following [Wilson], we learn point configurations to optimize inter-leaf distances, which serves as a proxy for learning evolutionary time, using Riemannian Stochastic Gradient Descent (RSGD).

We perform this optimization by updating the position of one leaf at a time, holding all other points fixed. Denote by $l(x_i)$ the log-likelihood as a function of the point x_i, corresponding to cell i only, holding all other points fixed. Denote by d_{x_j} the distance $d(x_i, x_j)$ between points x_i, x_j corresponding to cells (i, j).

Then, we can write:

$$l(x_i) = \sum_{j} \sum_{\text{sites } \sigma} \log \sum_{a \in A} \pi_a P_{a\sigma_i}(d_{x_j}/2) P_{a\sigma_j}(d_{x_j}/2)$$

In order to update the position of point x_i, we must derive a gradient update to perform RSGD.

For simplicity, consider first the expression:

$$g(x_i) = \log \sum_{a \in A} \pi_a P_{a\sigma_i}(d_{x_j}/2) P_{a\sigma_j}(d_{x_j}/2)$$

We will write $f(x_i) = \sum_{a \in A} \pi_a P_{a\sigma_i}(d_{x_j}/2) P_{a\sigma_j}(d_{x_j}/2)$ and denote $t = d_{x_j}/2$.

Then, we have

$$g'(x_i) = \frac{f'(x_i)}{f(x_i)}$$

By chain rule, we have:

$$f'(x_i) = f'(t) \times \frac{\delta t}{\delta x_i}$$

where $f(t) = \sum_{a \in A} \pi_a P_{a\sigma_i}(t) P_{a\sigma_j}(t)$

By the product rule, we have

$$\begin{aligned} f'(t) &= \sum_{a \in A} \pi_a P_{a\sigma_i}(t) Q_{a\sigma_j} P_{a\sigma_j}(t) + \pi_a P_{a\sigma_j}(t) Q_{a\sigma_i} P_{a\sigma_i}(t) \\ &= \sum_{a \in A} \pi_a P_{a\sigma_i}(t) P_{a\sigma_j}(t) \left[Q_{a\sigma_j} + Q_{a\sigma_i} \right] \end{aligned}$$

$$\frac{\delta t}{\delta x_i} = \frac{1}{2} \nabla_{x_i} d_{x_j}$$

All together, this gives:

$$g'(x_i) = \frac{\sum_{a\in A} \pi_a P_{a\sigma_i}(d_{x_j}/2) P_{a\sigma_j}(d_{x_j}/2) \left[Q_{a\sigma_j} + Q_{a\sigma_i}\right]}{2\sum_{a\in A} \pi_a P_{a\sigma_i}(d_{x_j}/2) P_{a\sigma_j}(d_{x_j}/2)} \times \nabla_{x_i} d_{x_j}$$

This yields a formula for the gradient of $l(x_i)$ as follows:

$$\nabla_{x_i} l(x_i) = \sum_j \sum_{\text{sites } \sigma} \frac{\sum_{a\in A} \pi_a P_{a\sigma_i}(d_{x_j}/2) P_{a\sigma_j}(d_{x_j}/2) \left[Q_{a\sigma_j} + Q_{a\sigma_i}\right]}{2\sum_{a\in A} \pi_a P_{a\sigma_i}(d_{x_j}/2) P_{a\sigma_j}(d_{x_j}/2)} \times \nabla_{x_i} d_{x_j} \tag{1}$$

We use the expression of the gradient of the distance function as derived by Wilson according to the Hyperboloid model of hyperbolic space.

For any $x, y \in H^m_\rho$, the gradient of the distance function is given by:

$$\nabla_x d_y = \frac{\rho^{-2} < x, y > x - y}{\sqrt{(\rho^{-1} < x, y >)^2 - \rho^2}} \tag{2}$$

1.3 Transition Probability Matrix

Inference under this model depends on the parameters of the transition probability matrix, P. These parameters can be informed by empirical knowledge of the behaviour of the system as determined by sensor screens performed *a priori* or can be inferred directly from observed data.

1.3.1 Structure of the Transition Matrix

For a given site, the transition probability matrix has a very defined structure imposed by the biological assumptions of the CRISPR system, and is characterised by the following terms. For each site the time taken to leave the base state and transition to either a mutated or deleted state follows an exponential distribution with rate $\lambda = \lambda_M + \lambda_D$, where λ_M corresponds to transitions to mutated stated and λ_D is a small value corresponding to transitions to a deleted state. Given that a transition occurs to some other character state, the transition to the observed character state, C, occurs with probability $p_{\phi C}$.

Once a transition to a mutated state has occurred, there can be no further transitions, except in the case of a deletion, which occurs at some low rate of λ_D. As deletions are expected to be rarely observed, we will assume that all states and all sites have equal rates of transition to a deletion.

Each site is, however, likely to have different mutation rates, λ_M as well as transition probabilities from base states to different characters.

For a sample alphabet of three potential states at two sites, the infinitesimal generator, Q, for the continuous time Markov Chain has the following structure:

	ϕ_1	ϕ_1	1	2	3	D
ϕ_1	$-(\lambda_{M1} + \lambda_D)$	0	$\lambda_{M1} p_{\phi 1}$	$\lambda_{M1} p_{\phi 2}$	$\lambda_{M1} p_{\phi 3}$	λ_D
ϕ_2	0	$-(\lambda_{M2} + \lambda_D)$	$\lambda_{M2} p_{\phi 1}$	$\lambda_{M2} p_{\phi 2}$	$\lambda_{M2} p_{\phi 3}$	λ_D
1	0	0	-λ_D	0	0	λ_D
2	0	0	0	-λ_D	0	λ_D
3	0	0	0	0	-λ_D	λ_D
D	0	0	0	0	0	0

where $\sum_{i\in\{1,2,3,\}} p_{\phi i} = 1$.

	ϕ_1	ϕ_1	M	D
ϕ_1	$-(\lambda_{M1} + \lambda_D)$	0	λ_{M1}	λ_D
ϕ_2	0	$-(\lambda_{M2} + \lambda_D)$	λ_{M2}	λ_D
M	0	0	-λ_D	λ_D
D	0	0	0	0

As several computations require computing the matrix exponential to determine transition probabilities, it will be helpful to express this matrix in a more compact form to take advantage of the structure of the transitions:

where M refers to any non-deleted mutated/character state.

1.3.2 Inferring Parameters of CRISPR/Cas9 System

Given a tree topology, the transition probability matrix $p_{0\to a}$ for a particular guide/site can be inferred by counting the number of branches along which such a transition occurred, and normalizing by the number of total branches. The mutation rate of that site/guide can be computed as the total number of branches on which any transition to non-zero state occurred, normalized by the total number of branches.

As such, we can use an initial tree to estimate the parameters of our system. Then, in an iterative manner, we can improve the inference of the tree topology given estimated parameters and use the improved tree to recompute updated model parameters.

Consider a given point i. Let β_j be the gradient of likelihood function with respect to point j (d_log_proba in the code). Let α_j be the denominator of the gradient of the distance function. $\sqrt{(\rho^{-1} < \text{points[i]}, \text{points[j]} >)^2 - \rho^2}$

At this point, the vector `self.gradients` contains: $\sum_j -\frac{\beta_j}{\alpha_j}\text{points[j]}$

As I understand, the function `project_onto_tangent_space` then modifies `self.gradients` as follows:

$$\begin{aligned}\text{scalar} &= \frac{1}{\rho^2} < \text{points[i]}, \sum_j -\frac{\beta_j}{\alpha_j}\text{points[j]} > \\ &= \frac{1}{\rho^2}\sum_j -\frac{\beta_j}{\alpha_j} < \text{points[i]}, \text{points[j]} >\end{aligned}$$

`self.gradients` $\leftarrow -\rho^{-2}\sum_j \frac{\beta_j}{\alpha_j} < \text{points[i]}, \text{points[j]} > \text{points[i]} +$ `self.gradients`

which yields:

`self.gradients` $\leftarrow -\rho^{-2}\sum_j \frac{\beta_j}{\alpha_j} < \text{points[i]}, \text{points[j]} > \text{points[i]} + \sum_j -\frac{\beta_j}{\alpha_j}\text{points[j]}$

`self.gradients` $\leftarrow \sum_j \beta_j[\frac{-\rho^{-2} < \text{points[i]}, \text{points[j]} > \text{points[i]} - \text{points[j]}}{\alpha_j}]$

`self.gradients` $\leftarrow \sum_j \beta_j[\frac{-\rho^{-2} < \text{points[i]}, \text{points[j]} > \text{points[i]} - \text{points[j]}}{\sqrt{(\rho^{-1} < \text{points[i]}, \text{points[j]} >)^2 - \rho^2}}]$